

REMARKS

Claims 1-19 are currently pending in this application.

As an initial matter, Applicant notes that the Examiner's initials are omitted for one of the references (reference "AR") on the PTO-1449 Form filed on December 22, 2000. The Examiner did sign the bottom of the PTO-1449 form as being considered on October 2, 2003. To the extent that reference "AR" has been considered, Applicant requests that the Examiner initial reference "AR" and provide Applicant's attorney with a copy of the fully initialed PTO-1449 Form.

Rejection of Claim 4 Under 35 U.S.C. §112

Claim 4 stands rejected under 35 U.S.C. §112 as failing to comply with the written description requirement, because a "Uniform Resource Locator" as recited in claim 4 was not described in the specification. Applicant amends the specification herein. The specification as amended now recites the term "Uniform Resource Locator". Support for the amendment can be found in claim 4 as originally-filed. No new matter has been added. Applicant respectfully requests that the Examiner reconsider the rejection and pass claim 4 to allowance.

Prior to discussing further the Examiner's rejection of the claims, we review the features of the present invention. Briefly, the present invention provides for automatically augmenting functionality in component-based application programming interfaces (API), and is particularly applicable to the Enterprise integration of software application programs. In general, an API receives a request from an application for functionality from a component. The component is queried regarding the requested functionality. If it is determined that the component does not support the requested functionality, a search for an augmentation component is initiated. If an appropriate augmentation component is found, it is dynamically loaded and used to respond to the requested functionality. Thus, the present invention provides for automatically augmenting functionality in an API to complete its at-the-moment "missing" functionality.

Base claims 1, 9, and 17-19 of the present application recite, in part, a method, apparatus, article of manufacture, and computer data signal embodied in a carrier wave for (i) receiving

from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality; (ii) querying the first component for the requested functionality; (iii) receiving from the first component that the requested functionality is not implemented by the first component; (iv) searching for an augmentation component to fulfill the requested functionality; (iv) loading the augmentation component to fulfill the request for functionality in place of the first component.

Rejection of Claims 1-3, 5-11, and 13-19 Under 35 U.S.C. §103(a)

Claims 1-3, 5-11, and 13-19 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,253,257 to Dundon (Dundon) in view of U.S. Patent No. 6,496,833 to Goldberg et al. (Goldberg et al.).

Turning briefly to the teachings of the cited references, the Dundon patent describes a software interface for dynamically mapping APIs to allow a single binary application to run in different environments without requiring different binary applications linked to different libraries. Dundon describes the utility within the context of a client-server network, in which different nodes of the network (e.g., workstations) may represent different environments. For example, a remote, limited-functionality node may represent a “lightweight” environment, versus a full-functionality node. Described in more detail at column 4, lines 49-62, the dynamic API mapping automatically detects libraries that it should use, loads the necessary libraries, and binds those libraries to the appropriate API functions at run time. In particular, an application specifically calls a “special” function that performs the API mapping procedure. Notably, as described beginning at col. 4, line 66, this special function loads and links the necessary libraries before calling any APIs. From that point forward, the linked APIs can be called by the application.

The Goldberg et al. reference describes using a query object generator tool to allow a client to construct a query object without being familiar with the underlying database language. The query objects allow business logic implemented in a business object to access a database management system (DBMS) without knowledge of the DBMS schema or the query language. Thus, the particular query objects described by Goldberg et al. are database queries, generally transmitted from a client to a database management system (DBMS) server.

In order to reject a claim under 35 U.S.C. §103(a), the Office Action must first establish a prima facie case of obviousness. Establishing a prima facie case of obviousness requires that: (i) there must be some suggestion or motivation, either in the reference or in the knowledge generally available to one of ordinary skill in the art, to modify the reference; (ii) there must be a reasonable expectation of success; and (iii) the prior art reference must teach or suggest all of the claim limitations. In re Vaeck, 947 F2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). The Examiner states “Dundon does not teach querying.” The Examiner further states that “it would have been obvious to apply the teachings of Goldberg in order to retrieve the desired information, which meets the characteristics specified in the query.” Applicant responds accordingly.

The Dundon patent neither implies nor suggests (i) receiving from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality, and (ii) querying the first component for the requested functionality, as claimed. Rather, at col. 4, lines 60-67, Dundon describes that an “executed application 116, 134 calls a function that performs the dynamic API mapping procedure.” When this occurs, libraries required by the invoked API are identified and dynamically loaded into the application and bound to the associated API function call. Notably, the required libraries are loaded and linked before any APIs are called. Thus, Dundon relies on the “special” function call to perform the necessary API mapping before any APIs are called. Essentially, Dundon needs to know what the requested functionality will be before any requests for the functionality are actually made. Only then will the necessary libraries be loaded and linked to support subsequent API calls. This is very different than performing API mapping in response to an API being called.

Likewise, Goldberg et al. fails to add these claimed features to Dundon. Rather, Goldberg et al. describes a tool for translating database queries. Querying a database is very different than querying the first component for the requested functionality, as claimed. A database query is directed to a DBMS to obtain a desired result with an expectation that the result will be obtained from the DBMS. Thus, the underlying functionality enabling the DBMS to respond to the query must already be in place. If it isn’t, then Goldberg et al. will translate the query as required so that it can be properly interpreted by the DBMS. Further, it would not have been obvious to modify Goldberg et al. to query any of the features/aspects of Dundon (to provide the claimed query of the first component for the requested functionality in the present

invention), because there would be no expectation that the request would be properly interpreted by the DBMS. Further, neither reference suggests the use of Goldberg's database querying in the API mapping setting of Dundon.

Even if Goldberg et al and Dundon were combined, the present claimed invention would not result. Although Goldberg et al provides querying of databases during execution of a client application, Dundon's API mapping is before the calling of any API. As shown in Fig. 5 and the corresponding description (Specification page 10), the present invention claimed steps are during run-time of the API. This patentable distinction is recited at least in base Claims 1, 9, 17, 18 and 19 with language (or similar language)

“...an application programming interface module receiving from an application a request for functionality to be fulfilled by a first component...;
a query unit within the application programming interface module querying the first component...”

Thus, neither reference taken individually or collectively disclose or otherwise make obvious the claimed invention of base claims 1, 9, and 17-19. Therefore, Applicant respectfully requests the withdrawal of the rejections of claims 1, 9, and 17-19 under 35 U.S.C. §103(a). Claims 2-3 and 5-8 depend directly and/or indirectly from base claim 1 and therefore contain all the elements of base claim 1. Similarly, claims 10-11 and 13-16 depend directly and/or indirectly from base claim 9 and therefore contain all the elements of base claim 9. For at least the same reasons, claims 2-3, 5-8, 10-11, and 13-16 are also non-obvious by the cited references. Therefore, Applicant respectfully requests withdrawal of the rejection of claims 2-3, 5-8, 10-11, and 13-16.

Rejection of Claims 4 and 12 Under 35 U.S.C. §103(a)

Claims 4 and 12 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Dundon in view of Goldberg et al. in view of U.S. Patent No. 5,896,533 to Ramos et al. (Ramos et al.).

The Examiner relies upon Ramos et al. for disclosing a Uniform Resource Locator, stating that “it would have been obvious to apply the teaching of Ramos to Dundon in order to establish the proper connection once the protocol is identified and established.” As argued above

with respect to claims 1, 9, and 17-19, Dundon fails to imply or suggest (i) receiving from an application a request for functionality to be fulfilled by a first component implementing at least a portion of the functionality, and (ii) querying the first component for the requested functionality, as claimed. Likewise, Ramos et al. fails to add these claimed features to Dundon. Rather, Ramos et al. describes allowing an application running on a computer that conforms to the Object Linking and Embedding API to automatically become capable of accessing the World-Wide Web.

As claims 4 and 12 respectively depend either directly or indirectly from base claims 1 and 9, the foregoing arguments and claim limitations apply. Thus, neither reference taken individually or collectively disclose or otherwise make obvious the claimed invention of claims 4 and 12. Therefore, Applicant respectfully requests that the rejections of claims 4 and 12 be withdrawn.

CONCLUSION

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By 

Mary Lou Wakimura

Registration No. 31,804

Telephone: (978) 341-0036

Facsimile: (978) 341-0136

Concord, MA 01742-9133

Dated: 1/8/04